(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平6－103064

(43)公開日　平成６年(1994)４月15日

(51)Int.Cl.[5]		識別記号	庁内整理番号	Ｆ Ｉ		技術表示箇所
Ｇ０６Ｆ	9/34	３３０	9189－5B			
	9/315					
	9/355					
			9189－5B	Ｇ０６Ｆ	9/30	３４０　Ｄ
			9189－5B		9/36	３２０

審査請求　未請求　請求項の数11(全 22 頁)

(21)出願番号　特願平4－276665

(22)出願日　平成４年(1992)９月21日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　中原　茂
東京都青梅市今井2326番地　株式会社日立製作所デバイス開発センタ内

(74)代理人　弁理士　玉村　静世

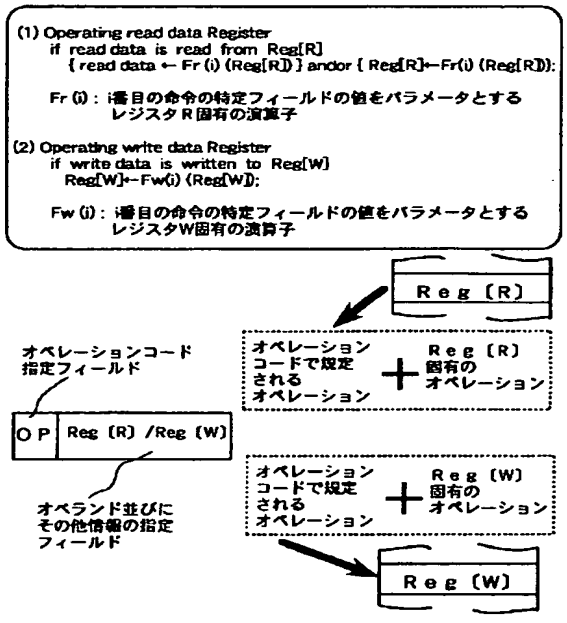
(54)【発明の名称】　データ処理装置及びそのデータ処理方法

(57)【要約】

【目的】　オペレーションコードのビット数や命令制御部の論理規模の増大を抑えつつ機能を多様に拡張できるデータ処理装置を提供することにある。

【構成】　レジスタに対するデータの書き込みまたは読み出し時に、そのデータに対して特定の処理を施す機能を予め所定のレジスタ（機能付きレジスタ）Ｒｅｇ〔Ｒ〕，Ｒｅｇ〔Ｗ〕に割当て、その機能付きレジスタを命令中にで指定することにより、そのデータに対して、命令のオペレーションコードによって規定される命令の処理と機能付きレジスタの有する特定機能とを組み合わせた多様なデータ処理を実現するものである。

【図１】



【特許請求の範囲】
【請求項１】　命令制御部の制御に基づいて命令を実行部で実行するデータ処理装置において、前記実行部に単数若しくは複数の機能付きレジスタを設けたものであって、
前記機能付きレジスタは、命令中でその利用が指定され、それを指定する命令のオペレーションコードで指示される処理によって当該機能付きレジスタから読出された情報或は書き込まれるべき情報に対して、当該オペレーションコードで指定される処理とは別の特定の処理を付加する機能が予め割当てられたレジスタであることを特徴とするデータ処理装置。
【請求項２】　前記機能付きレジスタは、これに書き込むべきデータの各ビット値を反転して保持するノットレジスタであり、当該ノットレジスタの入力には反転回路が接続されて成るものである請求項１記載のデータ処理装置。
【請求項３】　前記機能付きレジスタは、それが保持する値の読み出しが指定されることによって、当該読み出しデータに所定の定数を加算した値を再度保持するインクリメントレジスタであり、前記加算すべき定数を選択して出力する選択回路と、選択回路の出力と前記インクリメントレジスタの出力を加算して当該インクリメントレジスタの入力に供給する加算器とを備えて成るものである請求項１記載のデータ処理装置。
【請求項４】　前記機能付きレジスタは、これを指定する命令の偏値が負ならば保持値をリードデータとして前記命令による処理に引渡し、また、偏値が０または正ならば保持値から偏値を引いたものをリードデータとして前記命令による処理に引渡し、且つ、リードデータを引渡した後に当該保持値に前記偏値を加算した値を次の保持値として更新するモディファイドレジスタである請求項１記載のデータ処理装置。
【請求項５】　前記機能付きレジスタは、書き込みデータに対してマスクパターンとの論理積が採られたものを保持するアライメントレジスタである請求項１記載のデータ処理装置。
【請求項６】　機能付きレジスタは、書き込むべきデータにプログラムステータスワードのキャリー又はボローの状態を表わすビットの値を加算したものを保持するキャリー／ボローレジスタである請求項１記載のデータ処理装置。
【請求項７】　前記機能付きレジスタは、保持値の値を所定ビットだけ所定方向にシフトさせた値を、そのリードデータとして命令の処理の引渡すシフテッドレジスタである請求項１記載のデータ処理装置。
【請求項８】　命令の記述に従って処理を行うデータ処理方法において、
前記命令の記述は、オペレーションコードと、オペレーションコードで指定された処理に利用すべきレジスタの指定情報とを含み、
その命令中で指定されたレジスタに予め割当てられている固有のオペレーションを、当該レジスタの保持値に対して行い、その結果を前記レジスタからの読み出しデータとしてオペレーションコードで指定された処理に引渡す処理を行うことを特徴とするデータ処理方法。
【請求項９】　前記オペレーションコードで指定された処理に引渡した後、その引渡したリードデータを当該レジスタに書き戻す処理を更に含む請求項８記載のデータ処理方法。
【請求項１０】　命令の記述にしたがって処理を行うデータ処理方法において、
前記命令の記述は、オペレーションコードと、オペレーションコードで指定された処理に利用すべきレジスタの指定情報とを含み、
その命令中で指定されたレジスタに予め割当てられている固有のオペレーションを、その命令のオペレーションコードによって当該レジスタへの書き込みが指定されたデータに対して行い、これを当該レジスタに書き込む処理を行うことを特徴とするデータ処理方法。
【請求項１１】　前記命令は、予め固有のオペレーションが割当てられている前記レジスタとそれ以外の汎用レジスタとを区別可能なビットフィールドを前記レジスタ指定情報のためのフィールドに有するものである請求項９又は１０記載のデータ処理方法。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、オペレーションコードの増大を抑えつつ高機能な処理を実現し、また高機能な処理を少ないステップ数を以って実現可能なデータ処理装置並びにデータ処理方法に係り、例えば、マイクロコンピュータのような論理集積回路装置に適用して有効な技術に関するものである。
【０００２】
【従来の技術】フォン・イノマン型などのデータ処理装置の基本的な構成は程んど同じであり、例えば、キャッシュメモリ等に記憶した一連の命令を順番に命令制御部に読み込み、その命令が何であるかを表わしているオペレーションコードと呼ばれるビット領域をデコードしてデータに施すべき処理を決定し、更にオペランドと呼ばれる被演算データを指定する領域をデコードしてデータを取り込んで、上述した処理をそのデータに対して行うものである。このようなデータ処理装置の一例としては、Ｍｉｐｓ（ミップス）社のＲ４０００チップ（日経エレクトロニクス，１９９１年１０月１４日日経ＰＢ社発行，Ｎｏ．４９６号）、Ｉｎｔｅｌ（インテル）社のｉ９６０チップ（日経エレクトロニクス，１９９０年１月８日日経ＰＢ社発行，Ｎｏ．４９０号）、及びＨＰ（ヒューレットパッカード）社のＰＡ－ＲＩＳＣチップ（日経マイクロデバイス，日経ＰＢ社発行の１９９０年

9月号）などを挙げることができる。

【０００３】また、データ処理装置のアーキテクチャーは、比較的簡単な命令セットを以って処理の高速化並びにハードウェアの簡素化を図ろうとするＲＩＳＣ（Ｒｅｄｕｃｅｄ Ｉｎｓｔｒｕｃｔｉｏｎ Ｓｅｔ Ｃｏｍｐｕｔｅｒ）的アーキテクチャーと、オブジェクト指向のアーキテクチャのような比較的複雑な命令セットを以って高機能化を目指すＣＩＳＣ（Ｃｏｍｐｌｅｘ ＩｎｓｔｒｕｃｔｉｏｎＳｅｔ Ｃｏｍｐｕｔｅｒ）的アーキテクチャに大別できる。現実に提供されている各種データ処理装置のアーキテクチャがＲＩＳＣかＣＩＳＣの何れであるかというような峻別は実際には難しいが、多様なアドレッシングモードや、一つの命令で複数の演算を行う複合命令などをＣＩＳＣ的な命令と考えることができる。

【０００４】

【発明が解決しようとする課題】しかしながら、前記ＲＩＳＣ的なデータ処理装置において、ＣＩＳＣ的なデータ処理装置が１命令で提供するような高機能な処理は、基本的で簡単な処理を実現する命令を複数組み合せて対処していた。このため、出現頻度は少ないかもしれないが、そのような処理を能率的に行うことができなかった。このとき、多様なアドレッシングモードや、一つの命令で複数の演算を行う複合命令など、ＣＩＳＣ的な命令を基本的な命令セットに追加しようとすると、ハードウェアがサポートする命令数が増加してしまう。その結果、命令制御部のデコーダ等のランダムロジックの論理規模が大きくなり、クリティカルパスの遅延時間の増大、すなわちＬＳＩチップの性能低下やチップ面積の増加、開発日程の増大化などを招いてしまっていた。

【０００５】本発明の目的は、オペレーションコードの増大を抑えつつ機能の多様化を実現し、また高機能な処理を少ないステップ数を以って実現可能なデータ処理装置並びにそのデータ処理方法を提供することにある。今発明の別の目的は、ハードウェアの増大を極力抑えて多様な機能を実現できるデータ処理装置を提供することにある。

【０００６】本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【０００７】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【０００８】すなわち、レジスタに対するデータの書き込みまたは読み出し時に、そのデータに対して特定の処理を施す機能を予め所定のレジスタ（機能付きレジスタ）に割当て、その機能付きレジスタを命令中に指定することにより、そのデータに対して、命令のオペレーションコードによって規定される命令の処理と機能付きレジスタの有する特定機能とを組み合わせた多様なデータ処理を実現するものである。

【０００９】

【作用】上記した手段によれば、命令のオペレーションコードによって規定される命令の処理と機能付きレジスタの有する特定機能とを組み合わせて多様なデータ処理を実現できる。このことは、命令セットの命令数を増加することなしに、換言すれば、オペレーションコードのビット数を増大させることなく、データ処理装置による処理を多様化若しくは多機能化するように作用し、オペレーションコードのデコード論理に代表されるような命令制御部のハードウェアの増大を極力抑えて多様な機能を実現できる。

【００１０】上記によって実現される多様な機能は一つの命令中のオペランド指定領域などでの機能付きレジスタの指定で行われ、且つ、その機能付きレジスタに割当てれれる固有の機能は当該レジスタに対するデータの書き込みまたは読み出し時に行われるので、機能拡張に伴って実行すべき命令数が増えず、このことが、高機能な処理を少ないステップ数を以って実現するように作用する。

【００１１】

【実施例】図１には本発明に係るデータ処理装置の機能付きレジスタ（以下オペレーティングレジスタとも記す）の定義と機能の一例が示される。同図におけるオペレーティングレジスタの一般的定義はＣ言語的な形態で示され、オペレーティングレジスタは、（１）に示されるオペレーティング・リード・データ・レジスタ（Ｏｐｅｒａｔｉｎｇ Ｒｅａｄ Ｄａｔａ Ｒｅｇｉｓｔｅｒ）Ｒｅｇ〔Ｒ〕と、（２）に示されるオペレーティング・ライト・データ・レジスタ（Ｏｐｅｒａｔｉｎｇ Ｗｒｉｔｅ Ｄａｔａ Ｒｅｇｉｓｔｅｒ）Ｒｅｇ〔Ｗ〕に大別される。オペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕は、予じめ定義されているＲｅｇ〔Ｒ〕固有のオペレーションＦｒ（ｉ）を、ｉ番目の命令のＲｅｇ〔Ｒ〕で規定される特定のフィールドの値をパラメータとして、Ｒｅｇ〔Ｒ〕の内容に対して行い、その結果を読み出しデータ（ｒｅａｄ ｄａｔａ）とするか、再び当該Ｒｅｇ〔Ｒ〕に書き戻すかのどちらか一方か、またはその両方を行う。

【００１２】オペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕は、予じめ定義されているＲｅｇ〔Ｗ〕固有のオペレーションＦｗ（ｉ）を、ｉ番目の命令のＲｅｇ〔Ｗ〕で規定される特定のフィールドの値をパラメータとして、書き込みデータ（ｗｒｉｔｅ ｄａｔａ）に対して行い、当該Ｒｅｇ〔Ｗ〕に書き込む。

【００１３】前記オペレーティングレジスタＲｅｇ〔Ｒ〕，Ｒｅｇ〔Ｗ〕は、図１に示されるようなフォーマットの命令の中で指定することができる。図に示される命令は、オペレーションコード指定フィールドとオペ

ランド並びにその他情報の指定フィールドを有し、オペレーションコード指定フィールドには、データ処理装置のアーキテクチャに従って用意されているオペレーションコード（ＬＯＡＤやＡＤＤといった命令コード）ＯＰが記述され、オペランド並びにその他情報の指定フィールドには、前記オペレーションコードに従って命令を実行するために必要なオペランド（演算対象となるもの）それ自体やその所在、更にはその他制御情報が記述される。前記オペレーティングレジスタＲｅｇ〔Ｒ〕，Ｒｅｇ〔Ｗ〕は、命令フォーマット中において前記オペランド並びにその他情報の指定フィールドで指定される。その指定は、例えば、オペレーティングレジスタＲｅｇ〔Ｒ〕，Ｒｅｇ〔Ｗ〕に割当てられた固有のレジスタ番号で指定することができる。

【００１４】このように固有の機能が割当てられたオペレーティング・リード・レジスタＲｅｇ〔Ｒ〕を命令中で指定したとき、当該命令の実行においては、その命令のオペレーションコードで規定される処理（オペレーション）とオペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕固有の処理が、当該オペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕の保持情報を利用して行われる。例えば、その命令中で指定されたオペレーティングレジスタＲｅｇ〔Ｒ〕に予め割当てられている固有のオペレーションを、当該レジスタの保持値に対して行い、その結果を前記オペレーティングレジスタＲｅｇ〔Ｒ〕からの読み出しデータとしてオペレーションコードで指定された処理に引渡す処理を行う。前記オペレーションコードで指定された処理に引渡した後、その引渡した読み出しデータを当該オペレーティングレジスタＲｅｇ〔Ｒ〕に書き戻す処理を必要に応じて更に行うことができる。

【００１５】また、固有の機能が割当てられたオペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕を命令中で指定したとき、当該命令の実行においては、その命令のオペレーションコードで規定される処理とオペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕固有の処理が行われ、その結果が当該オペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕に格納される。例えば、その命令中で指定されたオペレーティングレジスタＲｅｇ〔Ｗ〕に予め割当てられている固有のオペレーションを、その命令のオペレーションコードによって当該オペレーティングレジスタＲｅｇ〔Ｗ〕への書き込みが指定されたデータに対して行い、これを当該オペレーティングレジスタＲｅｇ〔Ｗ〕に書き込む処理を行う。

【００１６】オペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕と、オペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕に割当てられた夫々固有の処理は、当該レジスタに対するデータの読み出し又は書込みに際して行われる。

【００１７】この原理的な説明から明らかなように、既存の所定オペレーションコードにオペレーティングレジスタＲｅｇ〔Ｒ〕，Ｒｅｇ〔Ｗ〕の指定を伴うことにより、オペレーションコードの種類を増やすことなく同一オペレーションコードすなわち１命令によって実現可能な処理を多様化若しくは豊富化でき、命令数並びに命令制御部の論理規模を増やすことなく高機能的な命令と同様の処理をサポートできるようになる。見方を変えれば、もともと高機能的な命令を専用のオペレーションコードで用意しているデータ処理装置においては、その機能を削減することなく命令数並びに命令制御部の論理規模を減らすことができる。

【００１８】次に図２から図８を参照しながら前記オペレーティングレジスタの具体例、並びにそれを用いた処理の一例をメモリ・レジスタ間のデータの流れに着目して概念的に説明する。

【００１９】図２にはノット・レジスタのＣ言語的な定義並びにそれを用いた処理例が概念的に示される。ノットレジスタＮは、前記オペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕の範疇に含まれ、ノットレジスタＮ（Ｒｅｇ〔Ｎ〕）への書き込みデータの各ビット値を反転させてノットレジスタＮ（Ｒｅｇ〔Ｎ〕）へ書き込むものである。同図には応用例としてディスティネーションにノット・レジスタＮを指定したＬＯＡＤ命令として「ＬＡＯＤ　ｄｉｓｐ(ｂ)，Ｎ」が示されている。この命令は、ディスティネーションにノットレジスタＮを指定することにより、汎用レジスタｂ｛ＧＲ〔ｂ〕｝の値（ベースアドレス）に偏値（ｄｉｓｐ）を加算したメモリアドレスに格納されているデータをノットレジスタＮに書き込むとき、当該ノットレジスタ固有の機能としてそのデータをビット反転する機能が付加される。したがって、同図に示される命令「ＬＡＯＤ　ｄｉｓｐ(ｂ)，Ｎ」のように、そのディスティネーションにノットレジスタＮを指定することにより、ノットレジスタＮへの書き込みデータをノットレジスタＮへ書き込む処理と共にその書き込みデータのビット反転を１命令（ＬＯＡＤ命令）で実行することができる。

【００２０】図３にはインクリメントレジスタのＣ言語的な定義並びにそれを用いた処理例が概念的に示される。インクリメントレジスタＩｎ（Ｒｅｇ〔Ｉｎ〕）は、前記オペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕の範疇に含まれ、インクリメントレジスタＩｎ（Ｒｅｇ〔Ｉｎ〕）のデータの読み出しが行われる際に、ある定数だけ、そのデータが示す内容に増加させて再びインクリメントレジスタＩｎ（Ｒｅｇ〔Ｉｎ〕）に書き戻すものである。同図の応用例では、インクリメントレジスタＩｎはＬＯＡＤ命令のベースアドレス指定用のレジスタとして使用される。このような応用例においては、一つのデータをメモリからレジスタに書き込みをする毎にインクリメントレジスタＩｎのベースアドレ

スが自動的に定数ずつ増加されるので、連続するメモリ領域のデータの読み出し処理を、通常のＬＯＡＤ命令を用いて実行できる。

【００２１】図４にはモディファイドレジスタのＣ言語的な定義並びにそれを用いた処理例が概念的に示される。モディファイドレジスタＭ（Ｒｅｇ〔Ｍ〕）は、前記オペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕の範疇に含まれ、ｉ番目の命令の偏値（ｄｉｓｐ）が負ならばモディファイドレジスタＭ（Ｒｅｇ〔Ｍ〕）が保持する値を読み出しデータ（ｒｅａｄ　ｄａｔａ）とし、偏値（ｄｉｓｐ）が０または正ならばモディファイドレジスタＭ（Ｒｅｇ〔Ｍ〕）の値からから偏値を引いたものを読み出しデータ（ｒｅａｄ　ｄａｔａ）とする。また、読み出しが行われる毎に、モディファイドレジスタＭ（Ｒｅｇ〔Ｍ〕）の内容に偏値を加算したデータを再びモディファイドレジスタＭ（Ｒｅｇ〔Ｍ〕）に書き戻す。このモディファイドレジスタＭ（Ｒｅｇ〔Ｍ〕）をＬＯＡＤ命令におけるベースアドレス指定用のレジスタとして使用した応用例が同図に示される。この例は、偏値（ｄｉｓｐ）が負の場合であり、モディファイドレジスタＭの定義における読み出しデータはモディファイドレジスタＭの値であるから、このときのメモリアクセスアドレスは、モディファイドレジスタＭの値に偏値（ｄｉｓｐ）を加算した値（Ｒｅｇ〔Ｍ〕＋ｄｉｓｐ）とされる。応用例における偏値（ｄｉｓｐ）が正の場合の具体例は図示していないが、モディファイドレジスタＭの定義における読み出しデータはモディファイドレジスタＭの値から偏値（ｄｉｓｐ）を引いたものとされるから、このときのメモリアクセスアドレスは、モディファイドレジスタＭの値に等しい値（Ｒｅｇ〔Ｍ〕－ｄｉｓｐ＋ｄｉｓｐ）になる。これにより、スタック若しくはＦｉＦｏ（Ｆｉｒｓｔ－ｉｎ　Ｆｉｒｓｔ－ｏｕｔ）のプッシュ／ポップのためのポストーインクリメント（Ｐｏｓｔ－Ｉｎｃｒｅｍｅｎｔ）とプリーデクリメント（Ｐｒｅ－Ｄｅｃｒｅｍｅｎｔ）のアドレッシングを、通常のＬＯＡＤ命令により実現することができるようになる。

【００２２】図５にはモディファイドインデックスレジスタのＣ言語的な定義が示される。モディファイドインデックスレジスタＭｉ（Ｒｅｇ〔Ｍｉ〕）は、図３のインクリメントレジスタＩｎ（Ｒｅｇ〔Ｉｎ〕）のオペレーションで、定数（Ｃｏｎｓｔ）の代わりにｉ番目の命令で指定されたレジスタＲｅｇ〔ｒ〕の値を使うものである。図３の応用例にモディファイドインデックスレジスタＭｉ（Ｒｅｇ〔Ｍｉ〕）を用いる場合、予じめＲｅｇ〔ｒ〕に適当な値を入れておくことにより、任意の値でベースアドレスの増加が行える。

【００２３】図５には更にアライメントレジスタのＣ言語的な定義並びにそれを用いた処理例が概念的に示される。アライメントレジスタＡＬ（Ｒｅｇ〔ＡＬ〕）は、前記オペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕の範疇に含まれ、インデックスで指定されるマスクパターンとアライメントレジスタＡＬ（Ｒｅｇ〔ＡＬ〕）への書き込みデータの各ビットごととの論理積を採るものであり、その論理積の処理をした後のデータをアライメントレジスタＡＬ（Ｒｅｇ〔ＡＬ〕）へ書き込む。例えば応用例１のようにマスクパターン（ＡＬ０）が上位ハーフワードは全て０、下位ハーフワードが全て１と定義されているものとすると、ＬＯＡＤ命令のディスティネーションとして指定されたアライメントレジスタＡＬ０には、下位ハーフワードにはメモリの対応するビットの値が、上位ハーフワードには全ビット０のデータが書き込まれる。このアライメントレジスタＡＬ０は図６に示される応用例２のように、ディスティネーションのレジスタｔにデータを書き込む際の中間バッファとして使用することもできる。図６においてその中間バッファはＡｇ０、マスクパターンは（Ａｇ０）として図示されている。したがって、応用例１，２に示すように、アライメントレジスタＡＬを書き込みデータのディスティネーション（又は中間バッファ）として指定することにより、その書き込みデータとマスクパターンとの論理積の処理とアライメントレジスタＡＬ（レジスタｔ）への書き込み処理とが１命令（ＬＯＡＤ命令）で実行できる。

【００２４】図７にはキャリー／ボロー（Ｃａｒｒｙ／Ｂｏｒｒｏｗ）レジスタのＣ言語的な定義並びにそれを用いた処理例が概念的に示される。キャリー／ボローレジスタＣ／Ｂ（Ｒｅｇ〔Ｃ／Ｂ〕）は、前記オペレーティング・ライト・データ・レジスタＲｅｇ〔Ｗ〕の範疇に含まれ、キャリー／ボローレジスタＣ／Ｂ（Ｒｅｇ〔Ｃ／Ｂ〕）への書き込みデータ（ｗｒｉｔｅ　ｄａｔａ）にプログラムステータスワードＰＳＷ（Ｐｒｏｇｒａｍ　Ｓｔａｔａｓ　Ｗｏｒｄ）のキャリーやボローの状態を表わすビット、すなわちＰＳＷ〔Ｃ／Ｂ〕の値を加算してキャリー／ボローレジスタＣ／Ｂ（Ｒｅｇ〔Ｃ／Ｂ〕）へ書き込むものである。。応用例のように加減算命令（ＡＤＤ命令等）のディスティネーションをキャリー／ボローレジスタＣ／Ｂに指定することにより、１命令の実行で、二つのレジスタａ，ｂの内容の加減算結果に、さらに前回の演算で発生したキャリーもしくはボローの値ＰＳＷ〔Ｃ／Ｂ〕を加えることができる。

【００２５】図８にはシフテッドレジスタＲｅｇ〔ｎｉ〕のＣ言語的な定義並びにそれを用いた処理例が概念的に示される。シフテッドレジスタｎｉ（Ｒｅｇ〔ｎｉ〕）は、前記オペレーティング・リード・データ・レジスタＲｅｇ〔Ｒ〕の範疇に含まれ、シフテッドレジスタｎｉ（Ｒｅｇ〔ｎｉ〕）の内容をｉビットだけ左側にシフトし、読み出しデータ（ｒｅａｄ　ｄａｔａ）として出力する。同図の応用例では、ＡＤＤ命令が示されていて、この命令を実行すると、シフテッドレジスタｎ２

から読み出されたデータは２ビット左シフトされた後にレジスタｂの値と加算され、その加算結果のデータがディスティネーションのレジスタｔに書き込まれる。

【００２６】図９には図２から図８で説明した機能付きレジスタを用いたデータ処理装置の一実施例ブロック図が示される。同図に示されるデータ処理装置は、公知の半導体集積回路製造技術によってシリコンのような１個の半導体基板に形成され、命令制御部１、演算部２、命令キャッシュメモリ３、及びデータキャッシュメモリ４が代表的に示される。

【００２７】同図の演算部２に示されるＥ３は、図２から図８で説明した各種機能付きレジスタの集合（以下単に機能付きレジスタとも記す）である。演算部２において、機能付きレジスタＥ３の機能実現に専用化された回路ブロックとして、特に制限されないが、セレクタＥ１、加算器Ｅ２、及び反転回路Ｅ４が新たに設けられる。マスク回路Ｅ５、セレクタＥ６、及びプリシフタＥ７は、データ処理装置には通常設けられていると共に前記機能付きレジスタＥ３の機能実現にも利用される回路ブロックである。演算部２にはその他に、複数本のレジスタから成る汎用レジスタＥ８、プログラムカウンタ（ＰＣ）Ｅ１０、算術論理演算器（ＡＬＵ）Ｅ９、及び前回の演算で発生したキャリー又はボローの値ＰＳＷ〔Ｃ／Ｂ〕等を保持しているプロセッサステータスワードレジスタ（ＰＳＷ）Ｅ１１などが設けられている。Ｂ１，Ｂ２，Ｂ３，Ｂ４は演算部２において代表的に示された内部バスであり、ＤＶ０～ＤＶ９はバイパス用ドライバである。また、機能レジスタ内のアライメントレジスタは、ＡＬ０の他にＡＬ１があり、それぞれのアライメントレジスタに対応してマスク回路のマスクパターンが後述する制御信号Ｓ６により制御される。

【００２８】命令制御部１は命令キャッシュメモリ３からフェッチした命令を解読して、代表的に図示された各種の制御信号Ｓ１乃至Ｓ１７などの制御信号を所定のタイミングを以って演算部２などに供給する。命令制御部１には、オペランドデコーダＩ１、オペコードデコーダＩ２、ターゲットデコーダＩ３、及び制御ブロックＩ４，Ｉ５が含まれる。オペコードデコーダＩ２は命令に含まれるオペレーションコードを解読する。命令のその他のフィールドは、特に制限されないが、オペレーションコードの種類に従って、オペランドデコーダＩ１やターゲットデコーダＩ３に供給される。オペランドデコーダＩ１は例えば命令のオペランド若しくはソース指定領域を解読する。この領域に機能付きレジスタとしての前記オペレーション・リード・データレジスタＲｅｇ〔Ｒ〕が指定されている場合には当該レジスタの指定が解読される。ターゲットデコーダＩ３は例えば命令のディスティネーション若しくはターゲット指定領域を解読する。この領域に機能付きレジスタとしての前記オペレーション・ライト・データレジスタＲｅｇ〔Ｗ〕が指定されている場合には当該レジスタの指定が解読される。オペランドデコーダＩ１及びターゲットデコーダＩ３による解読結果は、汎用レジスタＥ８や機能付きレジスタＥ３の指定に利用される。また、オペランドデコーダＩ１及びターゲットデコーダＩ３による解読結果は、オペコードデコーダの解読結果と共に制御ブロックＩ４にも供給されて、セレクタＥ１，マスク回路Ｅ５，プリシフタＥ７の制御並びにキャリー／ボロー制御、そして、バイパス用ドライバＤＶ０～ＤＶ９の開閉制御などに利用される。

【００２９】ここで図１０を参照しながら汎用レジスタＥ８や機能付きレジスタＥ３の指定手法について説明する。双方のレジスタは共に、命令中におけるオペランド並びにその他の情報のための指定フィールドで指定される。実際に命令中のどの領域すなわち第何ビット目から第何ビット目迄を利用するかはオペレーションコードの種類によって予め規定されている。例えば図１０に示されるようにＬＡＯＤ命令が、オペコード（オペレーションコード）、オペランド、ディスティネーション、偏値（ｄｉｓｐ）の各フィールドを有するとき、オペランド又はディスティネーションの領域でレジスタが指定される。例えば汎用レジスタＥ８と機能付きレジスタＥ３に含まれるレジスタが全部で２"個あるときには、ｎビットの情報によって順番にレジスタ番号を指定することができる。また、図１０に示されるように、例えば、汎用レジスタＥ８が８個の汎用レジスタＧＥＲｅｇ１～ＧＥＲｅｇ８を含み、機能付きレジスタＥ３が７個の機能付きレジスタＯＰＲｅｇ１～ＯＰＲｅｇ７を含むとき、個々のレジスタには同図に示される４ビットＢ３，Ｂ２，Ｂ１，Ｂ０で規定される番号を割り当てることができる。このように規定した場合には、レジスタ指定情報Ｂ３～Ｂ０の最上位ビットＢ３の”０”は汎用レジスタＥ８を指定するビットとみなされ、レジスタ指定情報の最上位ビットＢ３の”１”は機能付きレジスタＥ３を指定するビットとみなされる。したがって、このようにレジスタ番号の割り付けを行った場合、命令フォーマット中のレジスタ指定領域には汎用レジスタを指定するのか或は機能付きレジスタを指定するのかを指定するビットフィールド（Ｂ３）が存在することになる。

【００３０】プログラムカウンタＥ１０で指定されたアドレスにある命令は、命令キャッシュメモリ３から読み出され、命令制御部１へ送られる。命令制御部１は命令中のオペコード及びオペランドなどをデコードし、そのデコード結果にしたがった各種制御信号などで演算部２の動作を制御する。

【００３１】図１１及び図１２には図２から図８に示される各種機能付きレジスタに割当てられているオペレーションを図９の構成で実行するときの制御形態がパイプライン形式で示される。同図に示されるパイプライン処理は、命令フェッチ、命令デコード、演算、メモリアク

セス、及びレジスタライト／メモリストアの５段のパイプラインステージから構成され、機能付きレジスタを指定した処理毎に、どのステージでどのような処理が行われるかが示されている。同図において実線矢印はデータの流れ、破線矢印はアサートされた制御信号又はアドレス信号の流れを意味する。そして信号に付された符号並びにブロック内の符号は図９に示される符号に対応している。図２乃至図８で説明した各種機能付きレジスタに割当てられているオペレーションの一例を図９の構成に即して説明していく。

【００３２】ＬＯＡＤ命令等の実行時に、命令デコードステージにおいて、ターゲットデコーダＩ３が、デコードしたレジスタ番号がノットレジスタＮの場合は、マスク回路Ｅ５を介して読み出されるデータキャッシュメモリ４の読み出しデータ又は内部バスＢ２（以下書き込みバスとも記す）に読み出されているデータのビット値が、レジスタライトメモリストアステージでビット値を反転させる反転回路Ｅ４により、反転させられレジスタ選択制御信号Ｓ７により選択されるノットレジスタＮに書き込まれる〔図１１（１），（２）〕。

【００３３】ＬＯＡＤ命令等の実行時に、命令デコードステージにおいて、ターゲットデコーダＩ３が、デコードしたレジスタ番号がアライメントレジスタＡＬ０又はＡＬ１の場合は、ターゲットデコーダＩ３のデコード値により指定されたアライメントレジスタＡＬ０又はＡＬ１に対応するマスク回路Ｅ５のマスクパターンが、上記デコード値に基づいて、制御ブロックＩ４から出力される制御信号Ｓ６により選択される〔図１２（１１）〕。そして、レジスタライトメモリストアステージにおいて、データキャッシュメモリ４の読み出しデータ、内部バスＢ４（以下アドレスバスとも記す）に読み出されているデータ又は内部バスＢ５（以下書き込みバスとも記す）に読み出されているデータと選択されたマスクパターンとが、マスク回路Ｅ５により論理積がとられ、その論理積が施されたデータ、すなわち、マスク処理（以下アライメント処理とも記す）が施されたデータが、レジスタ選択制御信号Ｓ７により選択されるアライメントレジスタＡＬ０又はＡＬ１に書き込まれる〔図１２（１３）〕。またアライメントレジスタＡＬ０又はＡＬ１を図６の応用例２のように中間バッファとして用いる場合、アライメントレジスタＡＬ０又はＡＬ１に書き込まれたデータを、さらに他のレジスタ、例えば汎用レジスタＥ８内の所定のレジスタ等に内部バスＢ１（以下ソースバスとも記す）・Ｂ２等を介して書き込む。

【００３４】オペランドデコーダＩ１がデコードしたレジスタ番号が機能付きレジスタＥ３に含まれるレジスタであって、そのレジスタが前記シフテッドレジスタ（ｎｉ）ならば、データは、ｉに対応するビット数だけデータを左シフトするように、プリシフタＥ７が制御信号Ｓ１により制御される。すなわち、図１２に示されるように、命令デコードステージにおいてオペランドデコードが行われて当該シフテッドレジスタｎｉの指定が解読されると、当該ステージにおいてレジスタ選択制御信号Ｓ７により選択される機能付きレジスタＥ３に含まれるシフテッドレジスタｎｉからデータが内部バスＢ１に読出される〔図１２（１４），（１５）〕。次の演算ステージでは、オペランドデコードによって得られた制御信号Ｓ１によってプリシフタＥ７が制御されて、前記読み出しデータに対するシフト動作が行われる。シフトされたデータの処理は、当該シフテッドレジスタを指定した命令のオペレーションコードに従って処理される〔図１２（１６）〕。図８の応用例の場合だと、シフト処理されたデータは、算術論理演算器Ｅ９に入力される。そして演算ステージで算術論理演算器Ｅ９により、汎用レジスタＥ８内のレジスタｂに格納されているデータと加算処理が施されてレジスタライトメモリストアステージで汎用レジスタＥ８内のレジスタｔに、その加算処理結果のデータが格納される〔図１２（１６），（１７）〕。

【００３５】また命令デコードステージでオペランドデコーダＩ１がデコードしたレジスタ番号が機能付きレジスタＥ３に含まれるインクリメントレジスタＩｎの場合、レジスタ選択制御信号Ｓ７により選択される当該レジスタから読み出されたデータは内部バスＢ１及びプリシフタＥ７を通って算術論理演算器Ｅ９へ送られると同時に、内部バスＢ１を通って加算器Ｅ２へ送られる〔図１１（３），（５）〕。この加算器Ｅ２の他方の入力には、オペランドデコーダＩ１で解読されたインクリメントレジスタＩｎの選択制御信号Ｓ３によってセレクタＥ１で選択された固定値＋１が供給され、これによって、演算ステージで前記インクリメントレジスタＩｎの読み出し値に１が加算される〔図１１（８）〕。そしてレジスタライトメモリストアステージで、その格納されたデータが、書き込みバスＢ５を通じて再びインクリメントレジスタＩｎへ書き戻される〔図１１（９）〕。インクリメントレジスタＩｎの値が次ステージで使用されるときは、制御ブロックＩ４でそれを検出し、バイパス用ドライバＤＶ０またはＤＶ１を、制御信号Ｓ４またはＳ５をアサートすることによりイネーブル状態（開状態）にして、加算器Ｅ２の加算結果のデータをソースバスＢ１に直接出力しておく〔図１（１０）〕。図３の応用例の場合、算術論理演算器Ｅ９には、インクリメントレジスタＩｎの読み出しデータの他にプリシフタＥ７を介して、偏値ｄｉｓｐが供給されている。算術論理演算器Ｅ９は、これを加算することにより、データキャッシュメモリ４のアドレス計算をする〔図１１（６）〕。その後のステージ〔図１１（４），（７）〕については、モディファイドレジスタＭの動作説明時に説明する。

【００３６】命令デコードステージでオペランドデコーダＩ１がデコードしたレジスタ番号が機能付きレジスタＥ３に含まれる前記モディファイドレジスタＭの場合、

選択制御信号Ｓ３がセレクタＥ１の出力として偏値ｄｉｓｐを選択して加算器Ｅ２の片方の入力に偏値ｄｉｓｐを供給し、レジスタ選択制御信号Ｓ７により選択される当該レジスタのデータが内部バスＢ１を介して加算器Ｅ２の他方の入力に供給される〔図１１（３），（５）〕。演算ステージで加算器Ｅ２は、モディファイドレジスタMのデータの値と偏値ｄｉｓｐとを加算する〔図１２（８）〕。そして、レジスタライトメモリストアステージで、その加算されたデータが、内部バスＢ５を通じて再びモディファイドレジスタMへ書き戻される〔図１２（９）〕。次ステージでモディファイドレジスタMの値が使用されるときは制御ブロックＩ４でそれを検出し、バイパス用ドライバＤＶ０またはＤＶ１を、制御信号Ｓ９またはＳ５によりイネーブル状態にして、加算器Ｅ２の出力値を直接ソースバスＢ１へ出力しておく〔図１１（１０）〕。また、モディファイドレジスタMから読み出されたデータは、内部バスＢ１を介してプリシフタＥ７へも送られている。このとき、制御ブロックＩ４は当該偏値ｄｉｓｐが負であるのか０又は正であるのかを判定し、さらにオペコードデコーダＩ２の解読結果に基づいて、オペレーションコードに規定される命令がＬＯＡＤ命令か否かを判定する。そして制御ブロックＩ４は、当該偏値ｄｉｓｐの判定結果と上記命令の判定結果に基づいて、制御信号Ｓ１をプリシフタＥ７に出力する。プリシフタＥ７は、算術論理演算器Ｅ９の片方の入力に、モディファイドレジスタMから読み出されたデータを供給する。さらに、上記命令の判定結果がＬＯＡＤ命令の場合、プリシフタＥ７は、制御信号Ｓ１に基づいて、偏値ｄｉｓｐが正又は０ならば０を負ならば偏値ｄｉｓｐを選択して、算術論理演算器Ｅ９の他方の入力に供給する。尚、オペコードデコーダＩ２がデコードした命令が、ＬＯＡＤ命令で、そしてオペランドデコーダＩ１がデコードしたレジスタ番号がモディファイドレジスタM以外の場合、プリ下Ｅ７は、制御信号Ｓ１に基づいて、アドレッシングの種類によってデータを選択し（インデックスアドレッシング時はソースバスＢ１のデータを、偏値アドレッシング時は偏値ｄｉｓｐを選択して）、算術論理演算器Ｅ９へ送る。演算ステージで、データキャッシュメモリ４のアドレスがモディファイドレジスタMから読み出されたデータと偏値ｄｉｓｐ又は０とを算術論理演算器Ｅ９が加算することにより計算される〔図１１（６）〕。このアドレス計算は、図４の応用例を実現することになる。すなわち算術論理演算器Ｅ９の出力の値は、モディファイドレジスタMの読み出されたデータをM´として示すと、偏値ｄｉｓｐが負の時”M´＋ｄｉｓｐ”で、正又は０の時”M´”であり、応用例のメモリアクセスアドレスと内容的には同一である。算術論理演算器Ｅ９で計算されたアドレスは、アドレス信号として、アドレスバスＢ４へ送られ、これにより、データキャッシュメモリ４がそのアドレスに格納されているデータを読み出しデータとして、マスク回路Ｅ５へ出力する〔図１１（４）〕。レジスタメモリストアステージで、マスク回路Ｅ５は制御信号Ｓ６により制御され、ターゲットデコーダＩ３のデコードにより得られた書き込み先が汎用レジスタＥ８やアライメントレジスタ以外のオペレーティングレジスタの場合、マスク回路Ｅ５は、アライメント処理を実行せずに、上記書き込み先のレジスタに上記読み出しデータを書き込む。また、書き込み先がアライメントレジスタＡＬ０ならば、マスク回路Ｅ５は、そのレジスタＡＬ０に対応したマスクパターンのアライメント処理を上記読み出しデータに対して行い、アライメントレジスタＡＬ０若しくはアライメントレジスタＡＬ０を中間バッファとしてその他のレジスタへデータを書き込む〔図１１（７）〕。

【００３７】オペランドデコーダＩ１がデコードしたレジスタ番号がモディファイドインデックスレジスタＭｉの場合、制御信号Ｓ３がセレクタＥ１の出力としてソースバスＢ１を選択することにより、モディファイドインデックスレジスタＭｉの値のデータと任意のレジスタの値のデータとの加算が演算ステージにおいて加算器Ｅ２で行われる。以後の動作は、インクリメントレジスタＩｎのときと同様である。

【００３８】命令デコードステージで算術演算時のデータの書き込み先としてターゲットデコーダＩ３のデコードしたレジスタ番号がキャリー／ボローレジスタＣ／Ｂを指定すると、制御ブロックＩ４がそれを検出して、制御信号Ｓ２により、算術論理演算器Ｅ９の最下位ビットのキャリー入力として前回の演算時のキャリー又はボローの値ＰＳＷ〔Ｃ／Ｂ〕のデータが選択される〔図１２（１８）〕。演算ステージで算術論理演算器Ｅ９は２つのデータとＰＳＷ〔Ｃ／Ｂ〕のデータを加算する〔図１２（１９）〕。その加算結果のデータは、内部バスＢ２を通してキャリー／ボローレジスタＣ／Ｂへ書き込まれる〔図１２（２０）〕。尚、データの書き込み先がキャリー／ボローレジスタＣ／Ｂ以外の場合は、セレクタＥ６の出力は０が選ばれる。

【００３９】図１３には前記マスク回路Ｅ５の一例が示される。尚、図１３中のＳ６ａ～Ｓ６ｃは、前記制御信号Ｓ６に含まれる制御信号であり、Ｇ０ａ～Ｇ２３ａ，Ｇ１ｂ～Ｇ２３ｂ及びＧ０ｃ～Ｇ２３ｃは、それぞれ制御信号図Ｓ６ａ～Ｓ６ｃにより開閉制御されるゲート回路である。図１３に示される例は、レジスタのビット数が最大３２ビットの場合であり、入力に対してマスクせずに出力する態様、入力の上位３バイト（ビット００～ビット２３）をマスクして出力する態様、入力に上位ハーフワード（ビット００からビット１６）をマスクして出力する態様を有し、各態様は制御信号Ｓ６ａ，Ｓ６ｂ，Ｓ６ｃによって択一的に選択される。すなわち、ビット２４～ビット３１は入力から出力に至るスルーの信号経路を有する。ビット００～ビット２３には、Ｅ５ａ

の領域に示されるように制御信号Ｓ６ａで開閉制御されるゲート回路Ｇ０ａ～Ｇ２３ａを介して入力を選択的に出力に伝達する信号経路と、Ｅ５ｂの領域に示されるように制御信号Ｓ６ｂで開閉制御されるゲート回路Ｇ０ｂ～Ｇ２３ｂを介して論理値”０”の信号ビットを選択的に出力に伝達する信号経路とが設けられる。ビット００～ビット２３には、Ｅ５ｃ－１の領域に示されるように制御信号Ｓ６ｃで開閉制御されるゲート回路Ｇ０ｃ～Ｇ１５ｃを介して論理値”０”の信号ビットを選択的に出力に伝達する信号経路及びＥ５ｃ－２の領域に示されるように制御信号Ｓ６で開閉制御されるゲート回路Ｇ１６ｃ～Ｇ２３ｃを介して入力を選択的に伝達する信号経路が設けられる。前記夫々のゲート回路は、特に制限されないが、クロックドインバータのような３ステート出力回路を含んで構成される。

【００４０】ターゲットレジスタとしてアライメントレジスタＡＬ０及びＡＬ１が指定されない時（通常時）は、制御信号Ｓ６ａのみアサートされ、Ｅ５ａに含まれるゲート回路Ｇ０ａ～Ｇ２３ａがオン状態にされ、入力信号はそのまま対応するビットへ出力される。ターゲットレジスタとして、下位１バイトのみ“１”であるマスクパターンを持つアライメントレジスタＡＬ０又はＡＬ１が選択された時は、制御信号Ｓ６ｂのみがアサートされ、Ｅ５ｂに含まれるゲート回路Ｇ０ｂ～Ｇ２３ｂがオン状態にされ、入力信号の下位１バイトが、対応するビットに出力され、その他の上位３バイトには、“０”が出力される。ターゲットレジスタとして下位ハーフワードのみ“１”であるマスクパターンを持つアライメントレジスタＡＬ１又はＡＬ０が選択された時は、制御信号Ｓ６ｃのみがアサートされ、Ｅ５ｃ－１及びＥ５ｃ－２に含まれるゲート回路Ｇ０ｃ～Ｇ２３ｃがオン状態にされ、入力信号の下位ハーフワードが、対応するビットに出力され、その他の上位ハーフワードには、“０”が出力される。

【００４１】図１４には図９に示されるデータ処理装置のチップ平面が概略的に示される。同図において３は命令キャッシュメモリ（ＣＣ）、４はデータキャッシュメモリ（ＤＣ）、２は演算部（ＥＵ）、１は命令制御部（ＩＵ）、９はメモリ制御用ランダム論理（ＭＵ）及びシステムバス制御部（ＰＵ）、６は命令用タグキャッシュメモリ（ＣＡ）、７はデータ用タグキャッシュメモリ（ＤＣ）、５は命令用アドレス変換バッファ（ＣＴ）、７はデータ用アドレス変換バッファ（ＤＴ）、１０は入出力部（Ｉ／Ｏ）である。

【００４２】図１５には機能付きレジスタをサポートしていないデータ処理装置と本発明に係る機能付きレジスタをサポートするデータ処理装置の、パイプラインステージで区切った動作フローチャートの一例が示される。機能付きレジスタをサポートしていない場合には、命令フェッチサイクルで図９の命令キャッシュメモリ３から命令をフェッチし、デコードサイクルでフェッチした命令のオペコードデコードとオペランドデコードを行い、オペランドデコードによって示されるレジスタの内容を汎用レジスタから算術論理演算器に送る。演算サイクルでは、オペコードデコードで指示された演算を、フェッチしたレジスタの内容に対して算術論理演算器が行う。指示された演算がＬＯＡＤもしくはＳＴＯＲＥ命令ならば、続くメモリアクセスサイクルで、アクセスするメモリのアドレスと（ＳＴＯＲＥ時は）書き込むデータをデータキャッシュメモリへ送る。そして、レジスタライト・メモリストアサイクルで、演算したデータを汎用レジスタまたはデータキャッシュメモリへ書き込む。命令がデータのアライメントを指示するものであれば、このサイクルでそれを行った後に書き込みを行う。

【００４３】機能付きレジスタをサポートする場合には、デコードサイクルでデコードしたオペランドが、インクリメントレジスタＩｎ、モディファイドレジスタＭ、インクリメントモディファイドレジスタＭｉのとき、次の演算ステージにてアドレスモディファイ・インクリメントを行う。また、デコードしたオペランドがシフテッドレジスタｎｉの場合、演算サイクルで、演算・メモリアドレス計算を行う際に、あらかじめｉビットだけデータをプリシフトする。

【００４４】データの書き込み先のレジスタとして、アライメントレジスタＡＬ０又はＡＬ１、ノットレジスタＮが指定されたときは、レジスタライト・メモリストアステージにてデータのマスクあるいはビット反転を行ってからデータ書き込みを行う。また書き込み先のレジスタがキャリー／ボローレジスタＣ／Ｂならば、演算ステージで演算を行う時に、キャリー／ボローの値を算術論理演算器Ｅ９の最下位ビットに入力する。

【００４５】上記実施例によれば以下の作用効果がある。

（１）本実施例に係る機能付きレジスタをサポートするデータ処理装置では、上述した多様なアドレッシングモードや複合命令等を、レジスタに機能を持たせることにより行う。即ち、命令数を増加させる代わりに機能付のレジスタを設け、オペコード領域のビット数の増加をオペランド領域とターゲット領域のビット数で吸収する。これに比べて、機能付きレジスタをサポートしていないデータ処理装置では、アドレッシングの方式を多様化したり幾つかの命令を組み合わせた複合命令を追加したりしてＣＩＳＣ的な機能を基本的な命令セットに持たせようとすると、命令数を増加することになり、命令フォーマット中、オペコードフィールドのビット数が増加してしまい、オペコードデコーダの論理規模が大きくなり、命令デコード時間の増加、即ちデータ処理装置の動作周波数を低下させてしまうことになる。この相違は、図１６からも明らかなように、本実施例データ処理装置では機能拡張に際してオペレーションコード領域のビット数

並びにオペコードデコーダの論理規模の増大は全くない。これに対して、機能付きレジスタをサポートしない従来方式ではオペレーションコード領域のビット数並びにオペコードデコーダの論理規模は著しく増大する。

（2）更に、基本命令としてのオペレーションコードとオペランド保持領域としての機能付きレジスタとの組み合わせにより多様な機能を実現できるので、一つの機能付きレジスタを設ければ複数の命令若しくはオペレーションコードと組み合わせて使用できることになり、複数の機能を実現できる。例えば、ロード・アンドインクリメントと、インクリメントと、ストア・アンド・インクリメントの機能を付加しようとするとき、従来方式では、それらに個々に対応した命令若しくはオペレーションコードを追加することになるが、本発明ではインクリメント・レジスタを１個追加するだけで済む。したがって、データ処理装置に追加した機能の数に比べて、追加すべき機能付きレジスタの数は比較的小数で済み、オペランド領域とターゲット領域のビット数の増加があまり大きくならない。したがって、機能付きレジスタによってデータ処理装置の機能を拡張しても、命令のオペランドやターゲットの指定領域のビット数、そしてオペランドデコーダ並びにターゲットデコーダの論理規模の増大を最小限に抑えることができる。

（3）図２から図８までのオペレーティングレジスタを用いて、ＰＡ－Ｒｉｓｃで定義される命令と同様な機能を実現した場合の命令数の削減を本発明者が試算したところ、例えば、全命令セット１２１命令のうち、前記各種機能付きレジスタを用いた効果により、２６命令を削減できる。これは元の命令数の２１％に相当する。

【００４６】以上本発明者によってなされた発明を実施例に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。例えば、機能付きレジスタの種類は図２から図８で説明したものに限定されず、その他の機能を実現するようにもできる。以上の説明では主として本発明者によってなされた発明をその背景となった利用分野である汎用レジスタ方式のデータ処理装置に適用した場合について説明したが、本発明はそれに限定されるものではなく、アキュムレータ方式のデータ処理装置などにも適用可能である。また、専用的な処理に専ら適用されるようなデータ処理装置の場合にも汎用レジスタを不要にすることができる。

【００４７】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【００４８】（1）レジスタに対するデータの書き込みまたは読み出し時に、そのデータに対して特定の処理を施す機能を予め所定のレジスタ（機能付きレジスタ）に割当て、その機能付きレジスタを命令中にで指定することにより、そのデータに対して、命令のオペレーションコードによって規定される命令の処理と機能付きレジスタの有する特定機能とを組み合わせた多様なデータ処理を実現することができる。換言すれば、命令数の増加を抑えて、多様なアドレッシングモードや複合命令による処理と同等の機能を実現できる。

（2）命令のオペレーションコードによって規定される命令の処理と機能付きレジスタの有する特定機能とを組み合わせて多様なデータ処理を実現することにより、命令セットの命令数を増加することなしに、換言すれば、オペレーションコードのビット数を増大させることなく、データ処理装置による処理を多様化若しくは多機能化することができ、オペレーションコードのデコード論理に代表されるような命令制御部のハードウェアの増大を極力抑えて多様な機能を実現できる。これにより更に、命令制御部の命令デコーダ等のランダムロジックの論理規模が大きくならずに済み、ＬＳＩの性能低下も防止できる。

（3）上記によって実現される多様な機能は一つの命令中のオペランド指定領域などでの機能付きレジスタの指定で行われ、且つ、その機能付きレジスタに割当てれれる固有の機能は当該レジスタに対するデータの書き込みまたは読み出し時に行われるので、機能拡張に伴って実行すべき命令数が増えず、このことは、高機能な処理を少ないステップ数を以って実現できることも意味する。

（4）基本命令としてのオペレーションコードとオペランド保持領域としての機能付きレジスタとの組み合わせにより多様な機能を実現できるので、一つの機能付きレジスタを設ければ複数の命令若しくはオペレーションコードと組み合わせて使用できることになり、複数の機能を実現できる。したがって、データ処理装置に追加した機能の数に比べて、追加すべき機能付きレジスタの数は比較的小数で済み、オペランド領域とターゲット領域のビット数の増加があまり大きくならない。これにより、機能付きレジスタによってデータ処理装置の機能を拡張しても、命令のオペランドやターゲットの指定領域のビット数、そしてオペランドデコーダ並びにターゲットデコーダの論理規模の増大を最小限に抑えることができる。

【図面の簡単な説明】

【図１】本発明に係るデータ処理装置の機能付きレジスタの定義と機能の一例説明図である。

【図２】機能付きレジスタの一例であるノット・レジスタのＣ言語的な定義並びにそれを用いた処理例を概念的に示す説明図である。

【図３】機能付きレジスタの一例であるインクリメントレジスタのＣ言語的な定義並びにそれを用いた処理例を概念的に示す説明図である。

【図４】機能付きレジスタの一例であるモディファイド

レジスタのＣ言語的な定義並びにそれを用いた処理例を概念的に示す説明図である。

【図５】機能付きレジスタの一例であるモディファイドインデックスレジスタ並びにアライメントレジスタのＣ言語的な定義並びにそれを用いた処理例を概念的に示す説明図である。

【図６】前記アライメントレジスタを用いた別の処理例を示す説明図である。

【図７】機能付きレジスタの一例であるキャリー／ボローレジスタのＣ言語的な定義並びにそれを用いた処理例を概念的に示す説明図である。

【図８】機能付きレジスタの一例であるシフテッドレジスタのＣ言語的な定義並びにそれを用いた処理例を概念的に示す説明図である。

【図９】図２から図８で説明した機能付きレジスタを用いたデータ処理装置の一実施例ブロック図である。

【図１０】機能付きレジスタ及び汎用レジスタの指定方式を示す一例説明図である。

【図１１】機能付きレジスタのオペレーションを図９の構成に即して実行するときの制御方式をパイプライン形式で示す一例説明図である。

【図１２】図１１とは別の機能付きレジスタのオペレーションを図９の構成に即して実行するときの制御方式をパイプライン形式で示す一例説明図である。

【図１３】マスク回路の一例回路図である。

【図１４】図９に示されるデータ処理装置のチップ平面図である。

【図１５】機能付きレジスタを有しないデータ処理装置と本発明に係るデータ処理装置の、パイプラインステージで区切った一例動作フローチャートである。

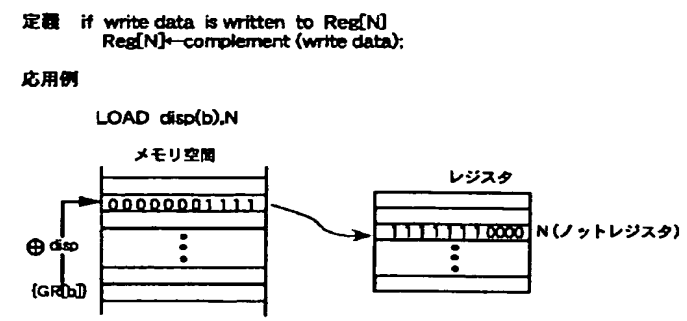
【図１６】命令のフォーマットとハードウェアとの関係を示した説明図である。

【符号の説明】

Ｎ　ノットレジスタ
Ｉｎ　インクリメントレジスタ
Ｍ　モディファイドレジスタ
Ｍｉ　モディファイドインデックスレジスタ
ＡＬ０　アライメントレジスタ
Ａｇ０　アライメントレジスタ（中間バッファ）
Ｃ／Ｂ　キャリー／ボローレジスタ
ｎｉ　シフテッドレジスタ
１　命令制御部
Ｉ１　オペランドデコーダ
Ｉ２　オペコードデコーダ
Ｉ３　ターゲットデコーダ
Ｉ４、Ｉ５　その他の制御ブロック
２　演算部
Ｅ１　セレクタ
Ｅ２　加算器
Ｅ３　機能付きレジスタ
Ｅ４　反転回路
Ｅ５　マスク回路
Ｅ６　セレクタ
Ｅ７　プリシフタ
Ｅ８　汎用レジスタ
Ｅ９　算術論理演算器
Ｅ１０　プログラムカウンタ
Ｂ３～Ｂ０　レジスタ指定情報
Ｂ３　汎用レジスタ／機能付きレジスタ指定ビット

【図２】

【図２】



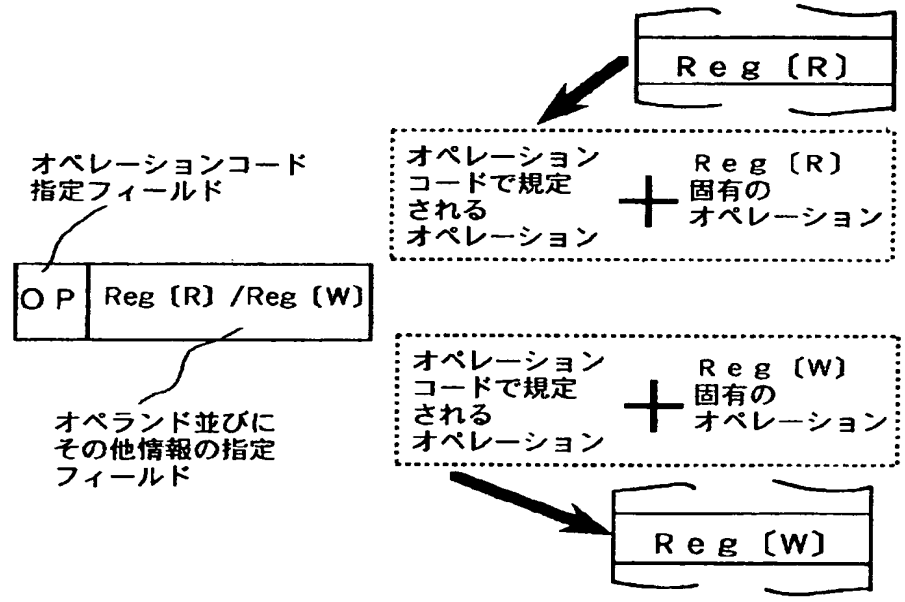
【図１】

(1) Operating read data Register
if read data is read from Reg[R]
{ read data ← Fr (i) (Reg[R]) } andor { Reg[R]←Fr(i) (Reg[R])};

Fr (i) : i番目の命令の特定フィールドの値をパラメータとする
レジスタＲ固有の演算子

(2) Operating write data Register
if write data is written to Reg[W]
Reg[W]←Fw(i) (Reg[W]);

Fw (i) : i番目の命令の特定フィールドの値をパラメータとする
レジスタW固有の演算子



【図３】

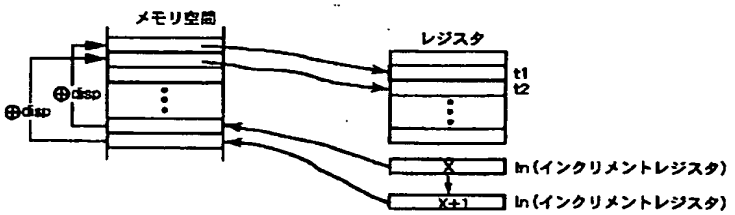
【図３】

2.インクリメントレジスタ

定義 if *read data* is read from Reg[In]
Reg[In]←Reg[In]+ Const.;

応用例

LOAD disp(In), t1
LOAD disp(In), t2



Const.=1 B

【図４】

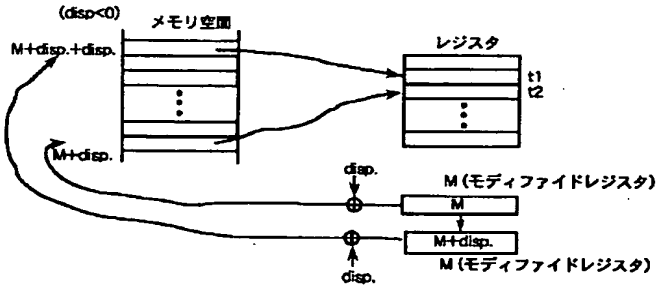
【図４】

3.モディファイドレジスタ

定義 if *read data* is read from Reg[M] {
if disp.(i) <0
read data ← Reg[M];
else
read data ← Reg[M] - disp.;
}
Reg[M] ←Reg[M] + disp.(i) ;

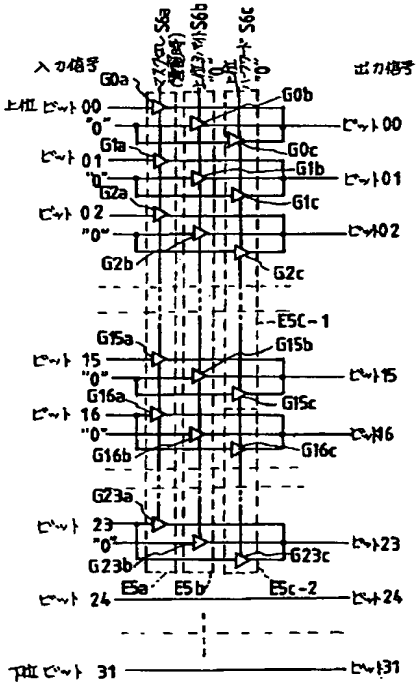
応用例

LOAD disp(M),t1
LOAD disp(M),t2

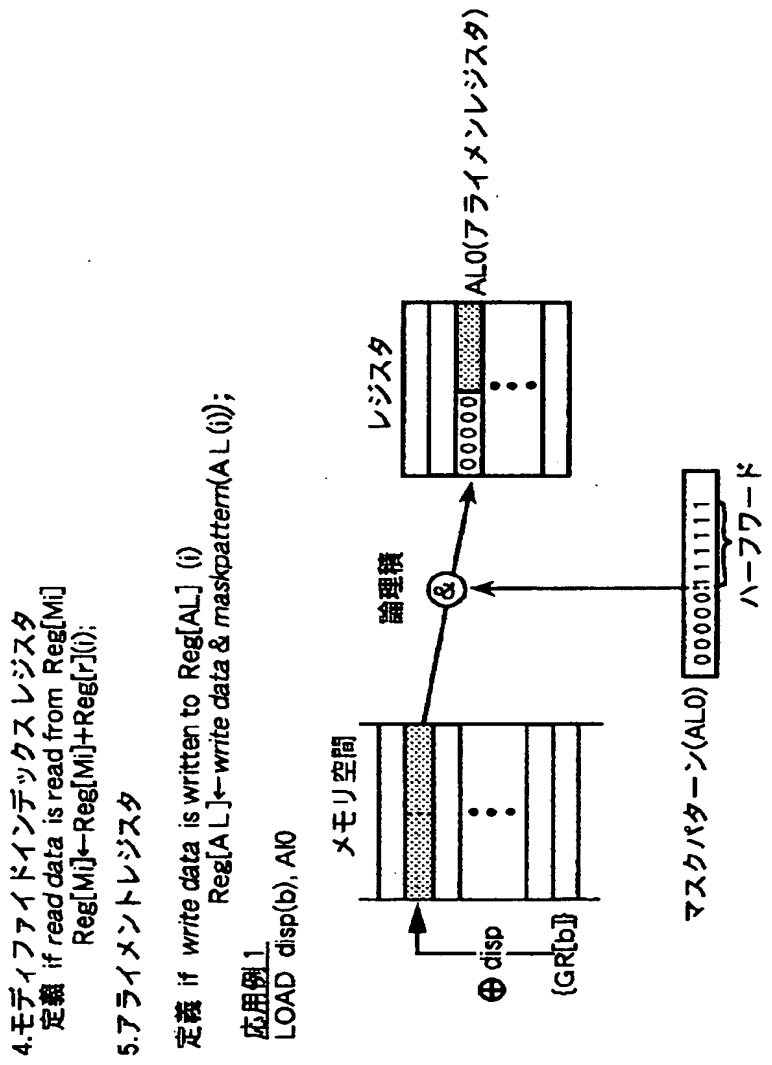


【図１３】

【図 13】

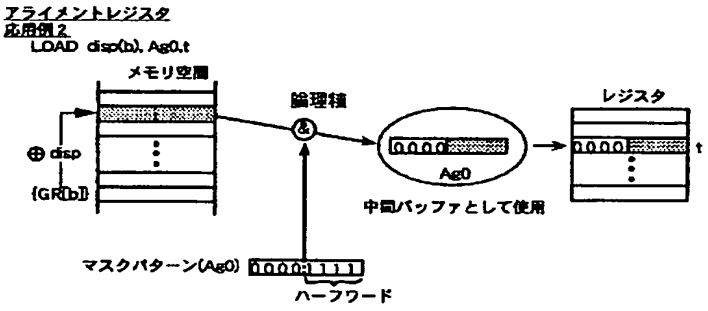


【図５】


4.モディファイドインデックス レジスタ
定義 if read data is read from Reg[Mi]
Reg[Mi]←Reg[Mi]+Reg[r](i);
5.アライメントレジスタ
定義 if write data is written to Reg[AL] (i)
Reg[AL]←write data & maskpattern(AL(i));
応用例1
LOAD disp(b), AI0
メモリ空間
⊕ disp
{GR[b]}
論理積
&
レジスタ
00000
AL0(アライメントレジスタ)
マスクパターン(AL0)
00000111111
ハーフワード

【図６】

【図６】



【図７】

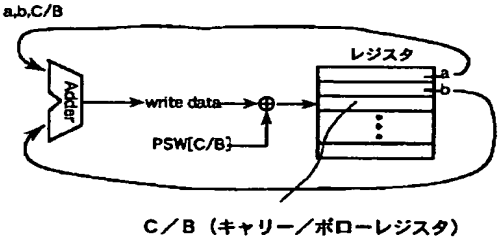
【図７】

6.キャリー／ボローレジスタ

定義　if write data is written to Reg[C/B]
Reg[C/B] ← write data + PSW[C/B];

応用例

ADD a,b,C/B



C／B（キャリー／ボローレジスタ）

【図８】

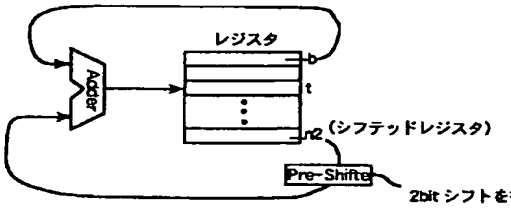
【図８】

7.シフテッド レジスタ

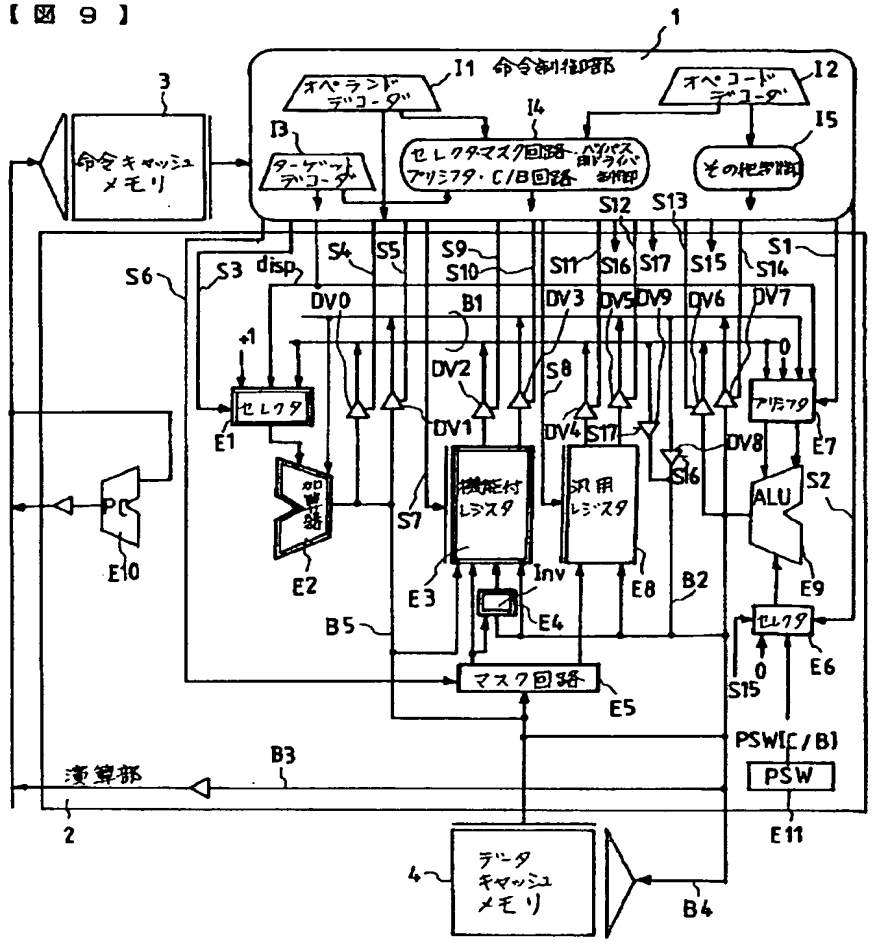
定義 if read data is read from Reg[ni]
read data ← shift (Reg[ni], i);

応用例

ADD n2,b,t

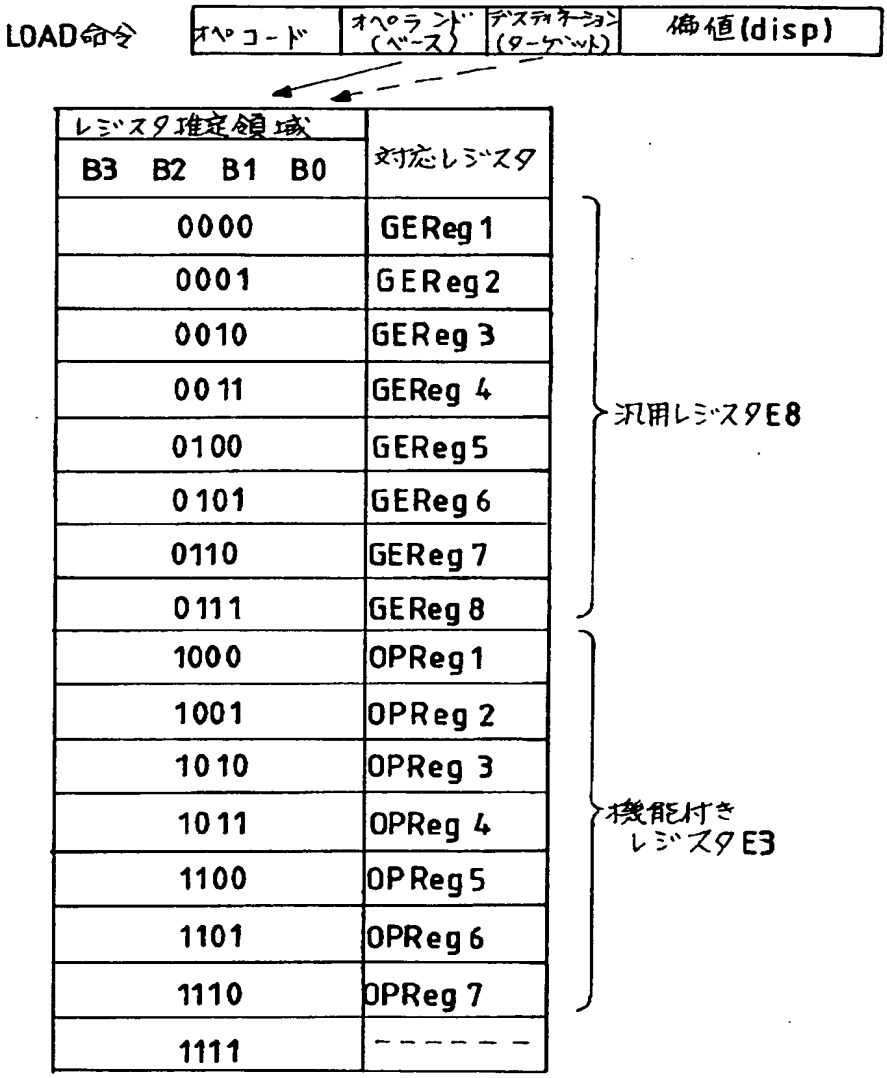


【図９】


【 図 ９ 】
1
I1 命令制御部
オペランドデコーダ
オペコードデコーダ
I2
I3
I4
I5
ターゲットデコーダ
セレクタマスク回路・バイパス・ドライバ・アリスタ・C/B回路
その他制御
3
命令キャッシュメモリ
S12
S13
S1
S3
disp
S4
S5
S6
S9
S10
S11
S16
S17
S15
S14
DV0
B1
DV3
DV5
DV9
DV6
DV7
+1
0
DV2
S8
セレクタ
E1
DV1
DV4
S17
DV8
S16
アリスタ
E7
ALU
S2
S7
E2
E3
E8
B2
E9
Inv
E4
B5
セレクタ
E6
0
S15
マスク回路
E5
PSW[C/B]
PSW
B3
演算部
2
E11
4
データキャッシュメモリ
B4

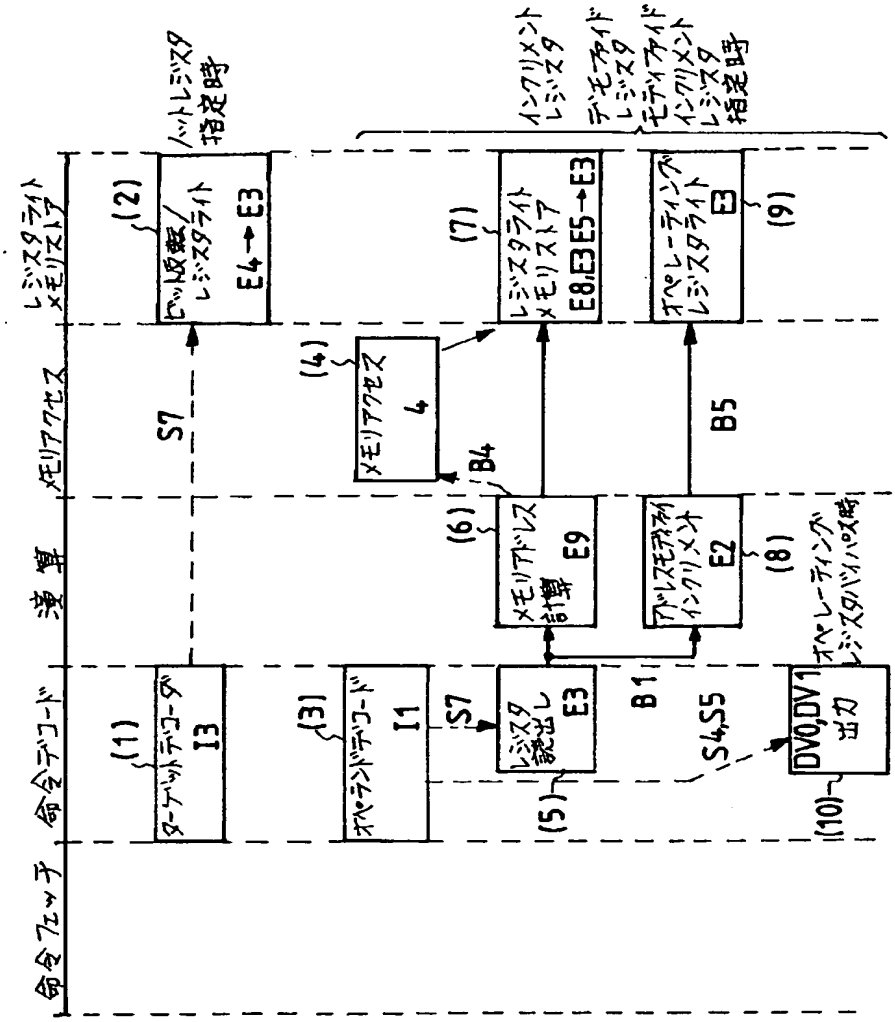
【図１０】

【図 10 】


LOAD命令
オペコード
オペランド（ベース）
デスティネーション（ターゲット）
価値(disp)
レジスタ確定領域
B3 B2 B1 B0
対応レジスタ
0000 GEReg1
0001 GEReg2
0010 GEReg3
0011 GEReg4
0100 GEReg5
0101 GEReg6
0110 GEReg7
0111 GEReg8
1000 OPReg1
1001 OPReg2
1010 OPReg3
1011 OPReg4
1100 OPReg5
1101 OPReg6
1110 OPReg7
1111 ------
汎用レジスタE8
機能付きレジスタE3

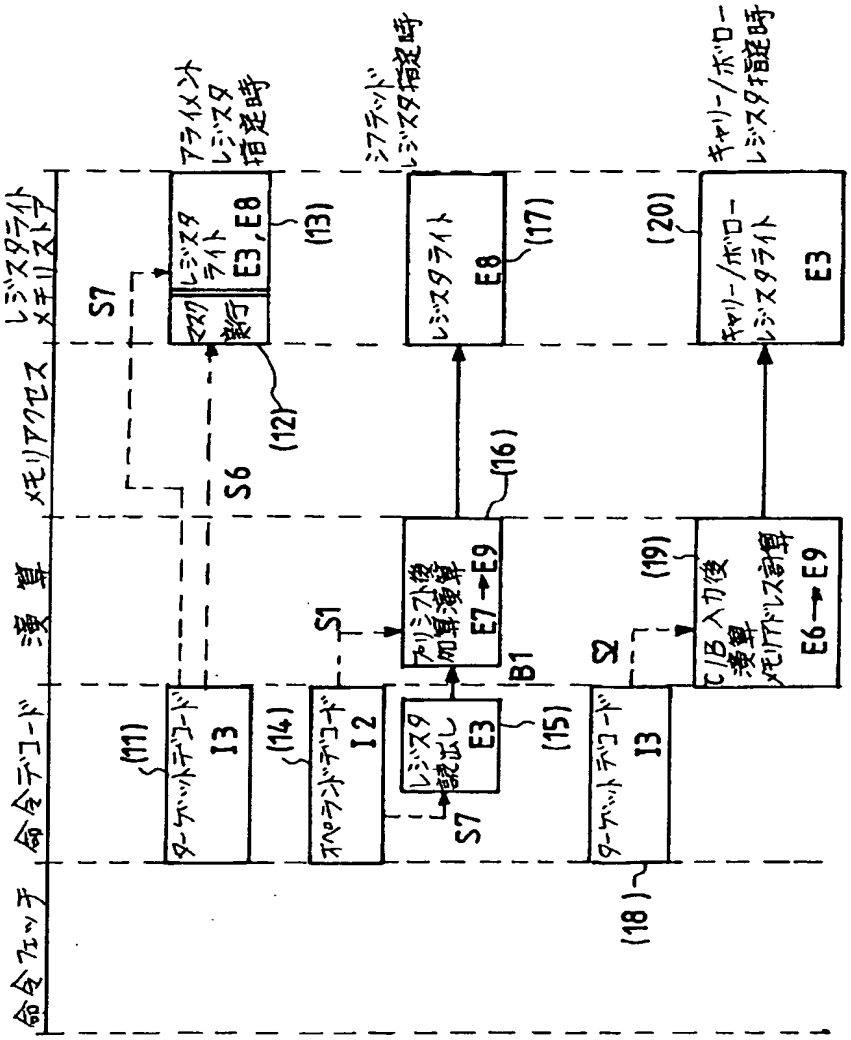
【図１１】

【図 11】


レジスタライト
メモリストア
メモリアクセス
演算
命令デコード
命令フェッチ
ハットレジスタ
指定時
インクリメント
レジスタ
デクリメント
モディファイ
インクリメント
レジスタ
指定時
(1)
ターゲットデコード
I3
(2)
ビット反転
レジスタライト
E4→E3
(3)
オペランドデコード
I1
(4)
メモリアクセス
4
(5)
レジスタ
読出し
E3
(6)
メモリアドレス
計算
E9
(7)
レジスタライト
メモリストア
E8,E3 E5→E3
(8)
アドレスモディファイ
インクリメント
E2
(9)
オペレーティング
レジスタライト
E3
(10)
DV0,DV1
出力
オペレーティング
レジスタバイパス時
S7
S7
B4
B5
B1
S4,S5

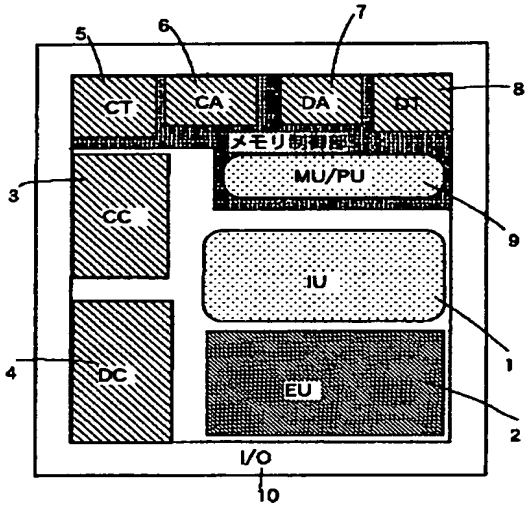
【図１２】

【図 12 】


命令フェッチ
命令デコード
演算
メモリアクセス
レジスタライト
メモリストア
ターゲットデコード I3
(11)
オペランドデコード I2
(14)
ターゲットデコード I3
(15)
(18)
レジスタ読出し E3
S7
S1
S2
B1
アドレス加算 E7→E9
(16)
C/B 入力後 演算 メモリアドレス計算 E6→E9
(19)
S7
S6
(12)
メモリ書込み 実行
レジスタライト E3, E8
(13)
アライメント レジスタ指定時
レジスタライト E8
(17)
シフティット レジスタ指定時
キャリー/ボロー レジスタライト E3
(20)
キャリー/ボロー レジスタ指定時

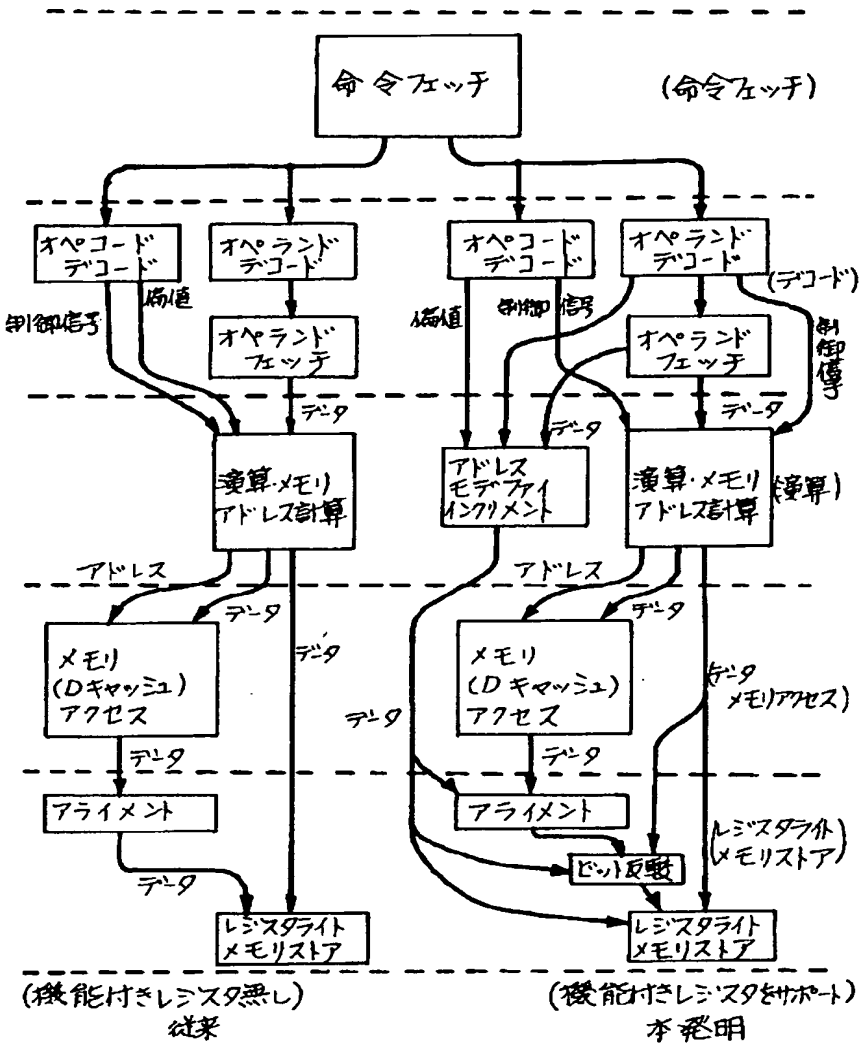
【図１４】

【図１４】

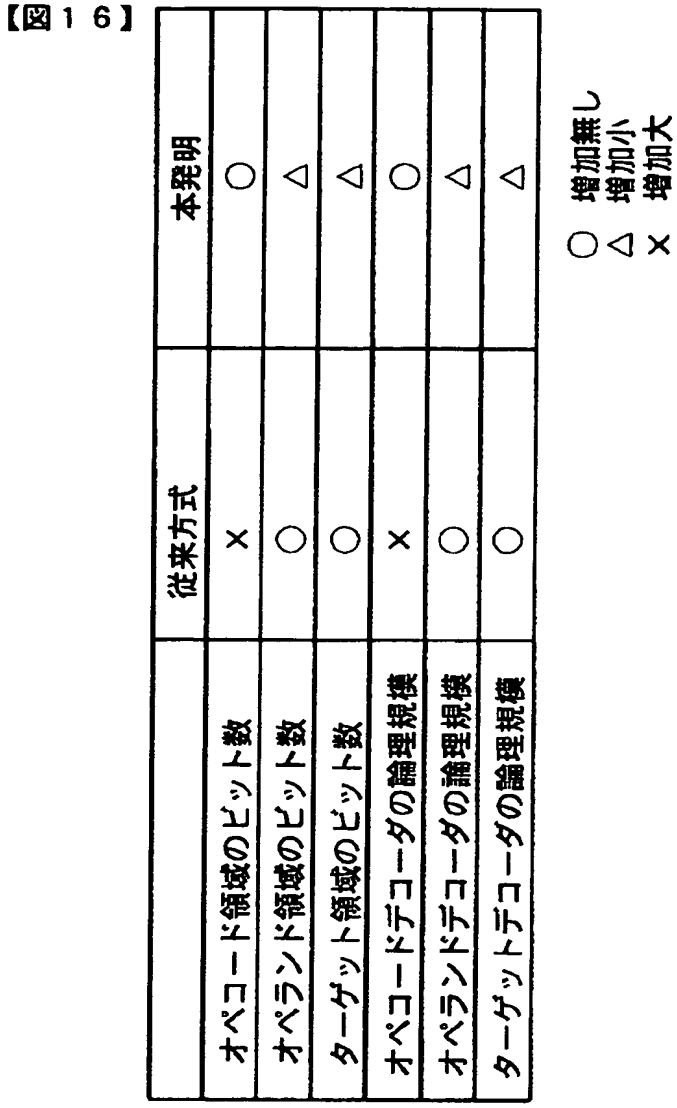


【図１５】

【図 15】


命令フェッチ
(命令フェッチ)
オペコード デコード
オペランド デコード
オペランド フェッチ
制御信号
値
データ
演算・メモリ アドレス計算
アドレス
メモリ (Dキャッシュ) アクセス
アライメント
レジスタライト メモリストア
(機能付きレジスタ無し)
従来
オペコード デコード
オペランド デコード
オペランド フェッチ
(デコード)
制御信号
値
データ
アドレス モディファイ インクリメント
演算・メモリ アドレス計算
(演算)
アドレス
メモリ (Dキャッシュ) アクセス
データ
メモリアクセス)
アライメント
ビット反転
レジスタライト メモリストア
(レジスタライト メモリストア)
(機能付きレジスタをサポート)
本発明

【図１６】



	従来方式	本発明
オペコード領域のビット数	×	○
オペランド領域のビット数	○	△
ターゲット領域のビット数	○	△
オペコードデコーダの論理規模	×	○
オペランドデコーダの論理規模	○	△
ターゲットデコーダの論理規模	○	△

○ 増加無し
△ 増加小
× 増加大